Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2016)
Ladislav Kavan and Chris Wojtan (Editors)

Versatile Interactions at Interfaces for SPH-Based Simulations

Tao Yang[†1], Ming Lin[1,2], Ralph Martin[3], Jian Chang[4], and Shi-Min Hu[1]

[1]Tsinghua University, [2]University of North Carolina at Chapel Hill, [3]Cardiff University, [4]Bournemouth University

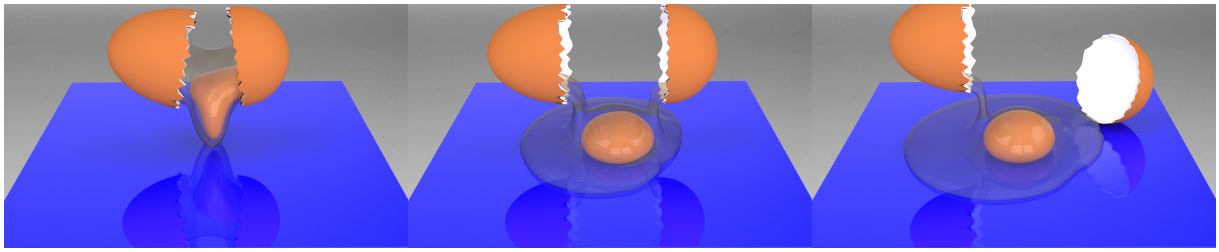

Figure 1: A simulated egg white and egg yolk flow out of the broken shell of a cracked egg, onto a plane, using our versatile approach to handling interactions at interfaces.

Abstract

The realistic capture of various interactions at interfaces is a challenging problem for SPH-based simulation. Previous works have mainly considered a single type of interaction, while real-world phenomena typically exhibit multiple interactions at different interfaces. For instance, when cracking an egg, there are simultaneous interactions between air, egg white, egg yolk, and the shell. To conveniently handle all interactions simultaneously in a single simulation, a versatile approach is critical. In this paper, we present a new approach to the surface tension model based on pairwise interaction forces; its basis is to use a larger number of neighboring particles. Our model is stable, conserves momentum, and furthermore, prevents the particle clustering problem which commonly occurs at the free surface. It can be applied to simultaneous interactions at multiple interfaces (e.g. fluid-solid and fluid-fluid). Our method is versatile, physically plausible and easy-to-implement. We also consider the close connection between droplets and bubbles, and show how to animate bubbles in air as droplets, with the help of a new surface particle detection method. Examples are provided to demonstrate the capabilities and effectiveness of our approach.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1. Introduction

Interactions involving interfaces between different phases of matter are commonly observed phenomena. For instance, the interaction between a fluid and air leads to surface tension, which is the main cause of many well known visual effects, including the water crown formed when a droplet falls into a liquid, and water jets. The interactions between a fluid and a solid are of two main kinds: fluid-solid coupling and adhesion. Fluid-solid coupling contributes to macroscopic movements while adhesion is caused by molecular forces and results in various wetting effects. Other interactions may also be observed at interfaces between different immiscible fluids. In daily life, many phenomena arise due to a variety of interactions at interfaces. For instance, when two bubbles collide in the air, interaction occurs between the gas inside,

† yangtao9009@gmail.com

the air outside, and the thin films around each bubble. To capture such phenomena, it is crucial to develop a versatile approach that can uniformly handle all types of inter-phase interactions.

Previous works mainly focus on one or two of these interactions at interfaces. Most rely on grid-based Eulerian simulators, modeling interfacial flow [LSSF06, Kim10, BB12, MEB*12, DBG14] or surface tension [TWGT10, BBB10, ZQC*14, DBWG15]. Smoothed particle hydrodynamics (SPH) has gained popularity as a particle-based method due to its mass-conservation and flexibility in handling topological changes. As many works show, SPH-based simulation is capable of simulating both surface tension [MCG03, BT07, SB12, AAT13, HWZ*14] and various other couplings [MSKG05, AIA*12, AAT13, BR15]. However, SPH methods are not without problems. For instance, the work addressing surface tension requires extra effort to handle other kinds of interactions at interfaces, such as adhesion, which restricts its application in a unified way to complex phenomena.

To capture interactions at interfaces, it is critical to accurately calculate the forces there. From the microscopic point of view, liquids, gases and solids are all composed of molecules. Physically, between any pair of molecules, the interaction force should be short-range repulsive and long-range attractive. Some work (e.g. [TM05, BT07]) has considered simulating surface tension using pairwise forces. However, recent works have criticized this approach due to its relatively poor performance, and have resorted to methods based on surface area minimization [AAT13], possibly in conjunction with air pressure [HWZ*14]. Although such methods achieve plausible results, surface area minimization is the result of surface tension, not its cause. Furthermore, surface energy minimization cannot be accurately implemented using SPH and leads to unsatisfactory results if extra forces are not taken into account. Air pressure is also not the main cause of surface tension.

In this paper, we use pairwise forces to simulate surface tension, as such an approach is easy to implement, cluster-preserving and momentum-conserving. In contrast to previous methods [TM05, BT07, AAT13] based on surface tension forces, we calculate pairwise forces using a *large* neighborhood of particles, as explained in Section 3. We go on to show that this pairwise view can be readily extended to handle a wide range of interactions. We introduce the idea of handling contact angles using pairwise forces from [TM05] into computer graphics, and expand it to deal with various kinds of interactions at solid interfaces. Our model is capable of simulating phenomena involving complex interactions at multiple interfaces using an SPH-based solver, e.g. cracking an egg (see Figure 1), and bubble animation (see Figure 9).

The main contributions of our work are:

- A new approach to refining pairwise forces, accurately capturing surface tension without extra forces or constraints.
- An expanded perspective of pairwise forces, capable of describing interactions at multiple interfaces in a versatile way.
- A method of animating bubbles in air as droplets for use in SPH-based simulations.

The paper continues with a discussion of related work in Section 2. We introduce our modified surface force model in Section 3. Methods for handling various interactions in a unified way are explained in Section 4. We demonstrate examples of our approach in Section 5, and finally, limitations and future work are discussed in Section 6.

2. Related Work

In computer graphics, interactions at interfaces between materials in different phases, or immiscible materials in the same phase, have been extensively investigated during the last decade. Work has considered such issues as surface tension, coupling, and interfacial flows. When using grid-based simulations, previous researchers mainly concentrated on interfacial flows [LSSF06, Kim10, BB12, MEB*12, DBG14], surface tension [TWGT10, BBB10, DBWG15] and bubbles [KLL*07, KLYK08, BDWR12, ZQC*14, DBWG15]. Since we focus on particle-based simulation, we do not discuss such work further .

SPH has proved its capabilities and effectiveness in handling such problems, and has been widely used in computer graphics. An important issue for SPH is tensile instability, which is the result of density underestimation at fluid-air and fluid-solid interfaces. Underestimated density values lead to negative pressures and particle clustering. An artificial pressure force was proposed by Monaghan [Mon00] to alleviate this problem, but this force causes spurious surface tensions. Another approach is to use higher order approximation of spatial derivatives in SPH discretization [ODAF07], but this increases the computation required. Akinci et al. [AIA*12] focused on fluid-solid interfaces and addressed the problem by precomputing a single layer of boundary particles with corrected volumes. Schechter and Bridson [SB12] dynamically sampled 'ghost' particles at both fluid-air and fluid-solid interfaces. He et al. [HWZ*14] instead used artificial air pressure forces to reduce computational costs. They achieved robust simulation of small-scale thin features in SPH by combining two-scale pressure estimation and anisotropic pressure filtering.

Researchers have developed a number of methods to model the interaction between a fluid and air. Early work focused on the continuum surface force (CSF) method [Mor00, MCG03], which aims to minimize surface curvature. This method relies on accurate curvature and normal estimates, which are unfortunately very sensitive to particle disorder. Because of the severe limitations of the CSF method, others [TM05, BT07] developed the pairwise interaction force

method, which works at the molecular level to overcome such problems. It uses a combination of short-range repulsion and long-range attraction forces to capture the physical nature of surface tension, avoiding erroneous estimates of curvatures and normals. However, as subsequent researchers [AAT13, HWZ*14] have noted, the pairwise forces used result in relatively poor performance. More recently, Akinci et al. [AAT13] simulated the surface tension force by combining a cohesion term and a surface area minimization term. This gives plausible results, but requires well-designed kernel functions, which cannot easily be extended to handle a variety of interactions at interfaces. He et al. [HWZ*14] drew inspiration from the Cahn-Hilliard equation and modeled surface tension by minimizing surface energy. However, multiple approximations are required to compute the surface energy, and as a result surface tension cannot be accurately captured.

Turning to interactions between a fluid and a solid, Akinci et al. [AIA*12] gave a versatile two-way fluid-solid coupling approach using SPH and per-particle correction of the volumes of boundary particles. Clavet et al. [CBP05] modeled the adhesion of fluids to solids using a distance-based attraction force. Schechter and Bridson [SB12] proposed the use of ghost particles and XSPH [Mon89] to capture realistic adhesion of a fluid to a solid. Following their work, He et al. [HWZ*14] calculated the air pressure force without sampling ghost air particles, with a significant reduction in memory and computational costs. By using a well-designed kernel function, Akinci et al. [AAT13] managed to capture different wetting effects.

Interactions between different fluids, however, have received less attention. Various particle-based multiple-fluid simulations [LLP11, RLY*14, YCR*15] have assigned different densities and labels to different fluids. Yang et al. [YCR*15] proposed a reactive stress term to capture interactions between miscible phases. Solenthaler and Pajarola [SP08] tried to handle fluids with large density differences more precisely. Müller et al. [MSKG05] noted the significance of fluid-fluid interactions and showed how to model dynamic phase changes and interfacial forces.

While many works have considered interactions at interfaces, none can simultaneously handle multiple phase interactions in a physically meaningful and computationally efficient manner. Our approach does so.

3. Surface Forces

3.1. Pairwise Interaction Term

We start by considering interactions between air and liquid, which result in surface tension. Physically, surface tension is the result of inter-molecular forces. Since SPH is a particle-based Lagrangian simulator, we may suppose some relationship exists between SPH particles and molecules; [TM05, BT07] have already attempted to simulate surface tension following this idea. They added a molecule-like pairwise particle-particle interaction term $\boldsymbol{F}^{interaction}$ into the conservative SPH approximation of the Navier-Stokes equation:

$$m_i \frac{D\boldsymbol{u}_i}{Dt} = \boldsymbol{F}_i + \boldsymbol{F}_i^{interaction}, \tag{1}$$

where m_i, $\boldsymbol{u}_i$ and $\boldsymbol{F}_i^{interaction}$ are the mass, velocity, and interaction term for particle i respectively. D/Dt is the substantial derivative corresponding to the Eulerian expression $\partial/\partial t + \boldsymbol{u} \cdot \nabla$. The term $\boldsymbol{F}_i$ combines all other forces acting on particle i and includes pressure forces, gravity, and so on. $\boldsymbol{F}_i^{interaction}$ is given by:

$$\boldsymbol{F}_i^{interaction} = \sum_j \boldsymbol{f}_{ji}, \tag{2}$$

where $\boldsymbol{f}_{ji}$ is the pairwise interaction force with which particle j acts on particle i; note that $\boldsymbol{f}_{ji} = -\boldsymbol{f}_{ij}$.

However, as previously noted, this method does not work well; Akinci et al. [AAT13] demonstrated that it is impossible to produce realistic surface tension effects by applying pairwise cohesion forces using only a finite support radius. They also claimed that pairwise cohesion forces cannot guarantee surface area minimization. Instead, they simulated surface tension by combining a pairwise cohesion force and a surface area minimization term; the former acts to alleviate particle clustering. However, it is physically implausible to claim that surface area minimization cannot be guaranteed by interaction forces. At the microscopic scale, interaction forces are the only forces existing between molecules. Since a free surface does not have fluid molecules on both sides, and the interaction force between fluid and air molecules is much lower, liquid molecules in such regions are pulled back into the liquid. It is this process that is responsible for minimizing the surface area. Thus, unlike Akinci et al. [AAT13], we focus on producing surface tension using pairwise interaction forces without additional constraints.

When SPH is used in computational physics, the number of neighboring particles in 3D is typically set to 80 to obtain the desired accuracy [TTP*14]. However, in computer graphics, the number of neighboring particles is set to about 30 for 3D simulation [IOS*14, Hoe12] in an attempt to balance realism and computational cost. However, our experiments have shown that doing so compromises the accuracy of the pairwise interaction forces in surface tension simulation. Increasing the smoothing radius leads to better estimates of pairwise surface tension forces.

3.2. Refined Surface Tension Model

Tartakovsky and Meakin [TM05] proposed use of a cosine function to generate pairwise interaction forces with a support radius h. As shown in Figure 2(a), starting from an initial configuration in the form of a cube (and ignoring external forces), their method fails to completely pull the particles

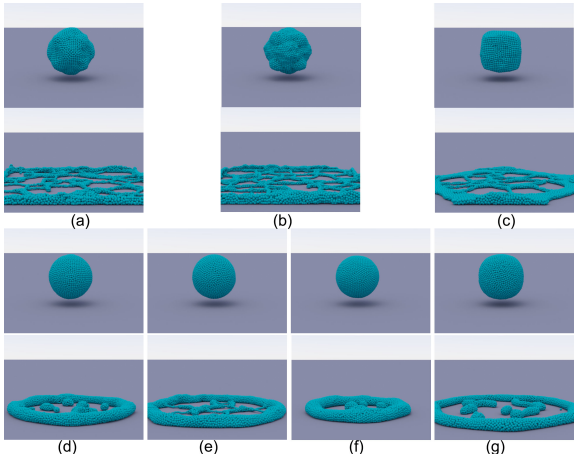

Figure 2: Results produced by surface tension forces using different modeling approaches; external forces are ignored. Top of each subfigure: configuration reached from a cube. Bottom: result of dropping that configuration onto a plane. Methods used are: (a) [TM05]; (b) [LMH06]; (c) [AAT13] without surface area minimization; (d) our refinement of [TM05]; (e) our refinement of [LMH06]; (f) our refinement of [AAT13] without surface area minimization; (g) [AAT13] with surface area minimization.

together into a sphere; the resulting configuration generates a cobweb-like series of elongated structures when dropped onto the plane. Akinci et al. [AAT13] were the first to note this problem, and solved it with the help of a surface area minimization constraint. To overcome the problem that the accuracy of the result is compromised if insufficient neighboring particles are used, we simply increase their number when considering pairwise forces instead of resorting to a surface area minimization constraint. There are two different ways this can be done: (i) by increasing the rest particle density for the whole fluid domain, or (ii) by enlarging the support radius for pairwise interaction forces. The former increases the computational cost tremendously. The latter, as our implementation demonstrates, achieves plausible results with relatively little extra computational load. The required enlargement ratio k for 3D simulation can be estimated to be:

$$k = (80/30)^{\frac{1}{3}} \approx 1.4. \tag{3}$$

We thus modify the original pairwise force in the work of Tartakovsky and Meakin [TM05] to:

$$\boldsymbol{f}_{ji} = \begin{cases} c_{ij} m_i m_j \cos\left(\frac{3\pi}{2kh}|\boldsymbol{r}_{ij}|\right)\frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}, & |\boldsymbol{r}_{ij}| \le kh \\ 0, & |\boldsymbol{r}_{ij}| > kh \end{cases}, \tag{4}$$

where $\boldsymbol{r}_{ij}$ is the vector between particle centres, and c_{ij} is a user-defined coefficient that controls the strength of the interaction force.

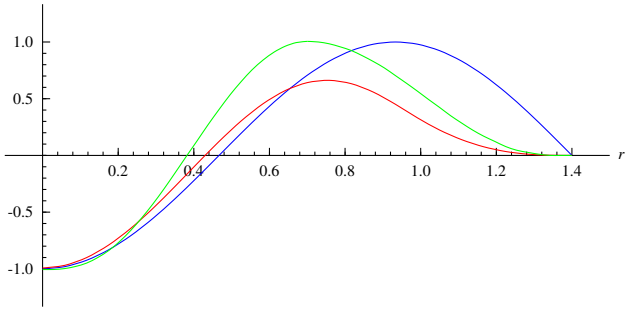

Figure 3: Interaction force models for surface tension, applying our refinement to the models of: Tartakovsky and Meakin's (blue), Liu et al. (red) and Akinci et al. (green). The forces are scaled to start at the same position. Our refinement to Liu's method. does not completely alleviate the cobweb-like structures in Figure 2(e), partly because repulsive and attractive magnitudes differ.

To examine the utility of the above idea, we have tested it using the surface tension model proposed by Liu et al. [LMH06] for dissipative particle dynamics (DPD) simulations, in which:

$$\boldsymbol{f}_{ji} = c_{ij} m_i m_j \big(AW(\boldsymbol{r}_{ij}, k_1 h) + BW(\boldsymbol{r}_{ij}, k_2 h)\big), \tag{5}$$

where A, B and k_1, k_2 are all user-determined parameters, and W is a kernel function. In particular, Liu et al. used a cubic spline kernel, while we use the Poly6 kernel function following Müller et al. [MCG03], to make it consistent with our SPH solver. Liu et al. set $k_1 = 1.0$ and $k_2 = 0.8$. However, as shown in Figure 2(b), this leads to poor performance. Again starting from a cube, the 3D result is far from a sphere and elongated cobweb-like structures appear on the plane. If we instead set $k_1 = 1.4$, $k_2 = 1.0$ following our idea above, the results are now an almost perfect sphere when starting from a cube, although cobweb-like elongated structures still appear in the plane (see Figure 2(e)).

We have also tested our idea using the kernel function proposed by Akinci et al. [AAT13] but enlarging the support radius. The results turn out to be plausible. This time we obtain an almost perfect sphere, without any cobweb-like structures (see Figure 2(f)).

These experiments show that enlarging the support radius for pairwise interaction forces is effective and results in more accurate surface tension. The cobweb-like elongating structures, discovered by Akinci et al. [AAT13], are caused by inaccurate surface tension forces. As demonstrated in Figure 2, the original pairwise interaction forces proposed by Tartakovsky and Meakin [TM05] and Akinci et al. [AAT13] both result in such unrealistic structures. With our refinement, the structures disappear. The functional form of the interaction force (see Figure 3) is also important, however. To avoid such cobweb-like structures, it seems that the repulsive

Algorithm 1 Simulation Loop

```
for all particles i do
    find neighboring particles
end for
for all fluid particles i do
    compute external forces
    initialize pressure and pressure force
end for
if bubble animation then
    detect surface particles (see Sections 4.2, 4.3)
end if
repeat
    for all fluid particles i do
        predict velocity and position
    end for
    for all fluid particles i do
        predict density
        update pressure
    end for
    for all fluid particles i do
        compute pressure force
        compute pairwise interaction force (see Section 3.2)
    end for
until enough iterations
```

and the attractive parts should have comparable magnitudes (see Figure 3).

Various ideas have been proposed to overcome neighborhood deficiency at the free surface, to alleviate the particle clustering which leads to tensile instability. These include the use of an additional artificial pressure [Mon00], 'ghost' air or solid particles [SB12], and per-particle volume correction [AIA*12]. Instead, we focus on the inaccurate surface tension forces caused by neighborhood deficiency and instead argue that an enlarged radius in the surface tension model can compensate for this deficiency. We have tested this basic idea on a variety of pairwise surface tension forces and the results indicate that this simple but powerful idea is plausible, and has promise for the graphics community.

The idea of using a different support radius is not completely new. For instance, He et al. [HWZ*14] adopted a two-scale pressure approximation to robustly estimate internal pressures for both water bodies and thin features. Physically, surface tension arises due to attraction between molecules, following the Lennard-Jones potential. For molecules inside the fluid, the attractions cancel one another out. For those near the surface, the asymmetrical distribution of neighbors leads to a non-zero net force towards the fluid domain. Adopting an enlarged support radius brings us closer to the physical nature of surface tension and leads to more realistic net force differences between interior molecules and those near the surface. Enlarging the support radius results in additional attractive forces, causing particle clustering in SPH. However, the nature of surface tension is to pull molecules together. The balance between surface tension and particle clustering is a trade-off. In our simulation, the pairwise forces we adopt do not vanish for close neighbors (see Figure 3), which helps to prevent particle clustering. Experimentally, we have observed that the enlargement ratio k used in our examples is capable of capturing accurate surface tension while simultaneously avoiding particle clustering.

Our method is easily implemented and can be integrated into state-of-the-art SPH solvers such as WCSPH [BT07] and PCISPH [SP09]. We first label every particle with a particular phase (see Section 4) and add our refined pairwise interaction forces according to the pairs of phases. When modeling surface tension using PCISPH, we take surface tension forces to be one of the internal forces, and add them to the pressure-correction iterations to ensure stable simulation. Our PCISPH based algorithm is outlined in Algorithm 1.

4. Interactions at Multiple Interfaces

With our refinement, the surface tension force can be accurately calculated from pairwise interaction forces without resorting to artificial considerations involving surface area minimization or air pressure. This approach has the further advantage of allowing us to extend the pairwise view to handle other types of interactions (e.g. fluid-solid and fluid-fluid) in a uniform manner. This is one of the biggest advantages of using pairwise interaction forces; it has so far been overlooked in computer graphics.

In this section, we first discuss the use of tuning coefficients for various interactions to achieve desired results. We then show how to animate bubbles in air using SPH, with the help of a newly introduced surface particle detection method.

4.1. Interaction Coefficients

To model surface tension, air and fluid both matter. Physically, air is also composed of molecules, so three different interaction forces between molecules must be considered when modeling surface tension: air-air, air-fluid, and fluid-fluid. Since the interaction between fluid and air is insignificant, we may idealize it as 0. While surface tension is actually a two-phase phenomenon, it can thus be modeled in terms of a single fluid phase. The interactions between pairs of air molecules do not impact the surface tension much, but play an important role in other ways, e.g. determining the contact angle.

The parameter c_{ij} used in Eqns (4) and (5) depends on the materials involved, and may be written as $c_{\alpha\alpha}$, where α represents a particular fluid phase. Bandara et al [BTO*13] realized that the coefficient c should vary according to the

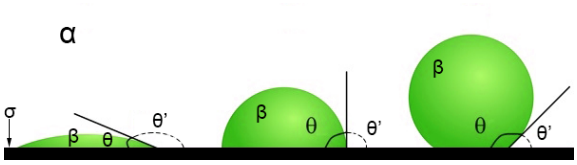

Figure 4: Different wetting effects and contact angles.

type of the interface. They focused on the adhesion of fluids to solids. By combining some classical theories of surface tension attributed to Young [You05], Maxwell [Max90], and Rayleigh [Ray64], they determined the relationship between contact angle and various coefficients to be:

$$\cos\theta = \frac{\bar{c}_{\alpha\alpha} - \bar{c}_{\beta\beta} + 2\bar{c}_{\sigma\alpha} - 2\bar{c}_{\sigma\beta}}{\bar{c}_{\alpha\alpha} + \bar{c}_{\beta\beta} - 2\bar{c}_{\alpha\beta}}, \tag{6}$$

where α, β are fluid phases and σ is the solid phase. Here, θ is the static contact angle between fluid β and the solid phase σ in the presence of fluid α, as illustrated in Figure 4; note the order of α and β matters. Often, α will be air. $\bar{c}_{\alpha\beta}$ is given by:

$$\bar{c}_{\alpha\beta} = n_\alpha n_\beta c_{\alpha\beta}, \tag{7}$$

where n_α is the number density of phase α. Eqn (6) is a precise mathematical model, but SPH does not require such accuracy. Furthermore, there are more coefficients in Eqn (6) than needed. For instance, when simulating the wetting effects of a fluid β on a solid σ, we only need to know the coefficients of $c_{\beta\beta}$ and $c_{\sigma\beta}$, as air particles are usually ignored in SPH. However, Eqn (6) requires coefficients associated with the air phase α, which makes Eqn (6) impractical for artistic control.

Thus, instead, we use Eqn (6) as a way to estimate the coefficients that actually control the simulation, and make approximations to make Eqn (6) more practical. Thus, if $\alpha, \beta, and \sigma$ represent air, a fluid, and a solid, respectively, the coefficients $c_{\alpha\beta}$ and $c_{\sigma\alpha}$ can be set to 0 because they are significantly smaller than the other coefficients in Eqn (6). While we can ignore interactions between air and liquid, interactions between particles in the same phase cannot be ignored, including those between air particles. The ignored air particles can in some ways be treated as ghost particles [SB12], so it is plausible to assume $n_\alpha = n_\beta$ and $c_{\alpha\alpha} = c_{\beta\beta}$. This simplifies Eqn (6) to:

$$\cos\theta = -\bar{c}_{\sigma\beta}/\bar{c}_{\beta\beta}. \tag{8}$$

To test Eqn (8), we choose α, β in an opposite way, i.e., letting α represent a fluid and β the air. Following the idea discussed above, Eqn (6) gives:

$$\cos\theta' = \bar{c}_{\sigma\alpha}/\bar{c}_{\alpha\alpha}, \tag{9}$$

where θ' represents the contact angle of a fluid α with solid σ in the presence of air β. Since $\theta, \theta' \in [0, \pi]$, combining Eqn (8) and Eqn (9), we see that, as illustrated in Figure 4,

$$\theta + \theta' = \pi. \tag{10}$$

For example, to simulate cherry sauce on a ball as considered later in Figure 13, since the contact angle of cherry sauce on the ball in air is close to 0, we use Eqns (8) and (9) and simply set $\bar{c}_{\sigma\alpha} = \bar{c}_{\alpha\alpha}$ according to Eqn (9) where α, β, σ represent cherry sauce, air, and the ball, respectively. Furthermore, if the cherry sauce and the ball are composed of particles with the same number densities, then $c_{\sigma\alpha} = c_{\alpha\alpha}$ according to Eqn (7). Eqns (8) and (9) are just simplified forms of Eqn (6) when one of the fluids is air.

When considering interactions at multiple interfaces, it is quite straightforward to sequentially determine appropriate coefficients for all interfaces. For instance, in the example of cracking an egg shown in Figure 1, we first determine coefficients for the interactions between air, egg white, and a solid using either Eqn (8) or Eqn (9). We then select coefficients for the interactions between egg white, egg yolk and a solid using the coefficients set previously, using Eqn (6), and so on. Such a sequence should always start at interactions involving air, since the interactions involving air are generally insignificant, allowing the simplified forms in Eqns (8) and (9)) to be used, simplifying the process of determining interaction coefficients.

Bandara et al [BTO*13] considered the particular topic of adhesion of fluids to solids to simulate different wetting effects. Contact angle is important, and cannot be ignored when handling interactions at solid interfaces. We use these ideas in computer graphics for the first time and further expand them to handle a wider range of interactions at interfaces. Since all materials are composed of molecules, one of the advantages of using pairwise interaction forces is that a uniform approach can be used without needing to design new kernel functions for different interfaces, e.g. as was necessary in Akinci et al. [AAT13]. By setting appropriate values for various coefficients according to the interaction being modeled, and using Eqn (6), our approach is capable of simulating complex phenomena with multiple interactions at different interfaces. See Figure 1.

4.2. Bubble Animation

Bubble simulation is an interesting and challenging topic. Bubbles can exist either in liquids or in the air (e.g. soap bubbles). SPH approaches have been used [MSKG05, CPPK07, KLYK08, IBAT11, PAKF13] to animate the movement of bubbles in liquids, but little has been done to simulate bubbles in air. The main difficulty lies in the thin structure of such bubbles. He et al. [HWZ*14] considered how to robustly simulate thin features using SPH, but their method is incapable of handling bubbles in air. We solve this problem by taking a different approach.

The shapes of bubbles in air are dominated by the sur-

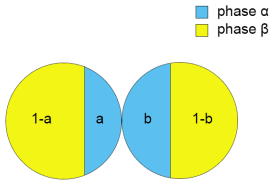

Figure 5: Interaction between two-phase particles

face tension of the thin film, as demonstrated by many grid-based methods [ZQC*14,DBWG15]. Droplets are also dominated by surface tension forces when external forces are absent. Both bubbles and droplets are volume-conserving at a constant temperature. Based on these considerations, we observe that it is possible to animate bubbles in the air by treating them as droplets.

An obvious difference between a bubble and a droplet is that they have different numbers of phases. A liquid droplet contains only a single fluid phase while a bubble has two different phases, a fluid phase on the surface and a gas phase inside. These two phases function differently. The gas phase dominates shape changes and conserves the volume of the bubble, while the fluid phase is responsible for various interactions. A straightforward way to simulate bubbles is to simply treat them as droplets without distinguishing the fluid and gas phases. However, we find that this approach makes the resulting bubbles look too rigid.

Instead, we consider each particle in the outermost layer of droplet to be composed of two phases, corresponding to a thin film in a fluid phase and an inner gas phase. We adopt the volume fraction model of [MSKG05] as a representation (for particles interior to the bubble, the fluid volume is considered to be 0). To simplify the implementation, we assume that the thin film of fluid is even over the whole bubble surface, so the volume fraction of every surface particle is the same. As the surface area changes along with the bubble shape, the volume fraction changes accordingly to conserve both fluid and interior gas volumes. A larger surface area leads to a thinner film and more surface particles, leading to a smaller volume fraction for the fluid phase in each surface particle.

In our approach, the surface particles are detected at every step in order to dynamically obtain the correct volume fraction. To determine the interaction forces between particles composed of two phases, the following interaction coefficient is used :

$$\tilde{c}_{ij} = abc_{\alpha\alpha} + \big(a(1-b)+(1-a)b\big)c_{\alpha\beta} + (1-a)(1-b)c_{\beta\beta}, \tag{11}$$

where particles i and j each contains both phases α and β, and a, and b are the volume fractions of phase α in particles i and j respectively (see Figure 5).

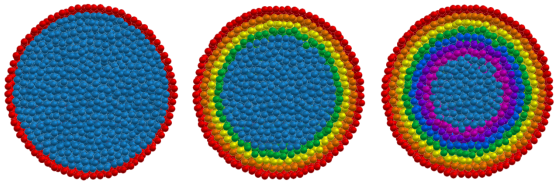

Figure 6: Results of surface particle detection. Surface particles can be accurately detected layer by layer (from left to right).

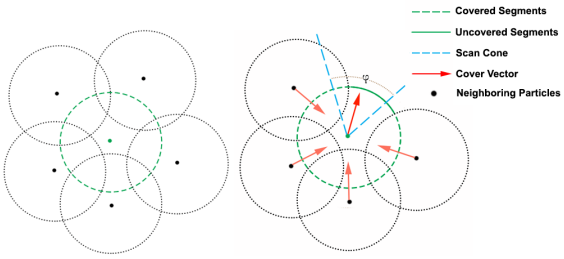

Figure 7: Surface particle detection.

When simulating bubbles as droplets, the surface particles are composed of a fluid phase and an inner-gas phase, while the others are considered as inner-gas particles. The fluid phase fraction of a surface particle actually varies from particle to particle; we set it to the same value for each particle for simplicity.

4.3. Surface Particle Detection

It is not easy to detect surface particles reliably and effectively in SPH. One approach is to label as surface particles those whose magnitude of smoothed *color gradient*, or whose number of neighboring particles, is larger or smaller than a threshold. Our experiments show that these methods are sensitive to particle disorder, and incapable of labeling just a single layer of surface particles as such. Solenthaler et al. [SZP07] proposed an alternative method to detect surface particles. Although it gives better results, they are still not completely satisfactory. We instead give an effective approach based on the work of Barecasco et al. [BTN13], which after modification, performs well (see Figure 6).

The i-th SPH particle and its neighboring particles are considered to be solid spheres of the same size, rather than points. The i-th particle is considered an interior particle only if its spherical surface is fully covered by the spherical surfaces of its neighbors (see Figure 7 left); surface particles have uncovered spherical surface segments (see Figure 7 right). If a particle has any uncovered spherical segments, it must be a surface particle. Barecasco et al. [BTN13] com-

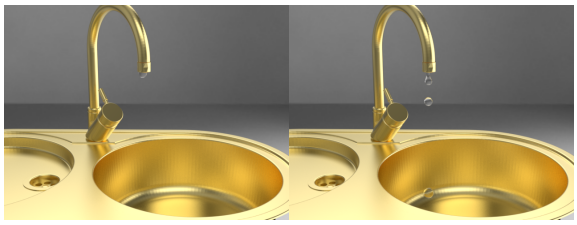

Figure 8: A dripping tap.

Table 1: Performance on Examples

Example	particles	steps/frame	msec/frame	frames/sec
Egg cracking	180k	3	30.0	33.3
Water splashing	232k	2	52.0	19.2
Cherry sauce	10k-40k	1	14.6	68.5
Wetting effects	9k	2	12.5	80.0
Water dripping	0k-7k	2	16.5	60.6
Blowing bubbles	0k-7k	2	25.6	39.1
Popping bubbles	41k	2	40.5	24.7

pute a cover vector $\boldsymbol{b}_i$ for particle i defined by:

$$\boldsymbol{b}_i = \sum_j \boldsymbol{r}_{ij}/|\boldsymbol{r}_{ij}|, \tag{12}$$

where j runs over neighboring particles within a search radius R_1, which leads to severe errors in handling the geometrical cavities [BTN13]. Ideally, $\boldsymbol{b}_i$ should point to the uncovered sphere segment, if any. To detect whether the indicated segment is covered or not, a scan cone is used (see Figure 7). Every neighboring particle is checked; the i-th particle is determined to be an interior one if any neighboring particle j within a search radius R_2 satisfies the equation below:

$$\arccos\left(\frac{\boldsymbol{r}_{ji}}{|\boldsymbol{r}_{ji}|}\cdot\frac{\boldsymbol{b}_i}{|\boldsymbol{b}_i|}\right) \leq \frac{\varphi}{2}, \tag{13}$$

where φ is the angle subtended by the scan cone. A suitable value depends on the particle distribution; in our experiments, we set $\varphi = \pi/2$.

Barecasco et al. [BTN13] set the search radii R_1 and R_2 to the support radius of the SPH kernel function, i.e. h. However, we find it works better to set R_1 slightly smaller than h as it is optimal to calculate the cover vector $\boldsymbol{b}$ using a single layer of neighboring particles. If an uneven distribution of neighboring particles contributes to the cover vector, its direction is only slightly affected due to the averaging process. R_2, on the other hand, should be set slightly larger than h in order to avoid some mis-detections due to particle disorder.

5. Results

5.1. Examples

We have implemented our algorithm using CUDA on an NVIDIA GeForce GTX980 GPU using single precision, as it is adequate for graphical simulations and reduces memory requirements. Our algorithm has been tested on simulating various challenging scenarios which we now describe. Performance for these examples is given in Table 1; we achieve real-time performance in all cases. The particle numbers shown include both fluid and boundary particles. Times presented are averages. The results were rendered using Mitsuba [Jak10].

Cracking an egg: Figure 1 illustrates that our method is capable of simulating complex phenomena with various types of interface interactions. A number of coefficients associated with interfaces between egg white, egg yolk, air and solid were sequentially set using Eqn (6). The egg white and egg yolk present different wetting effects. Letting α, β and σ represent egg white, egg yolk, and solid respectively: $c_{\alpha\alpha} = -6.0\times10^3, c_{\beta\beta} = -1.6\times10^4, c_{\sigma\alpha} = -8.0\times10^3, c_{\sigma\beta} = -2.0\times10^3$ and $c_{\alpha\beta} = 1.0\times10^3$.

Water crown: our surface tension model can be used to capture the details of water splashing, generating a crown and the 'i' pattern typically captured by photography. See Figure 12. Our refined method achieves similar results to those of Akinci et al. [AAT13] without the need for an extra surface area minimization term. We set the interaction coefficient to -1.6×10^4.

Cherry sauce on a ball: a stream of cherry sauce flows over a ball. This example shows that our approach can model the adhesion force between a fluid and a solid by setting appropriate coefficients for solid and fluid phases without requiring extra 'ghost' air particles or artificial air pressure forces. Here, $c_{\sigma\alpha}$ and $c_{\alpha\alpha}$ are both set to -1.2×10^4, where α and σ represent the fluid and solid phases respectively.

Popping bubbles: Figure 10 shows our method for simulating bubbles in air as droplets. On the left, a small cluster of bubbles settles into an equilibrium. On the right, the cluster rearranges itself into a new equilibrium after one bubble bursts.

Water jets and blowing bubbles: our method can be used to capture commonly observed phenomena, such as a dripping tap (see Figure 8). There is a strong connection between water jets and blowing bubbles; our approach is the first particle-based approach in graphics to be able to simulate blowing bubbles (see Figure 9).

5.2. Discussion

As shown in Algorithm 1, we merely detect surface particles when simulating air bubbles. To tune parameters, we first determine if interactions exist at solid interfaces, and then adopt different approaches accordingly. For instance, only one interaction coefficient needs tuning in examples such as the water crown and water jets (as no solids are involved),

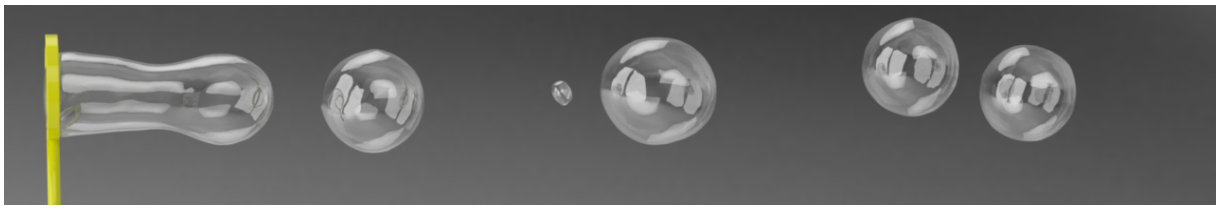

Figure 9: Blowing bubbles. Our method is capable of animating bubbles in the air as droplets. There is a close connection between blowing bubbles and water jets. Using this observation, we provide the first particle-based approach to simulate blowing bubbles.

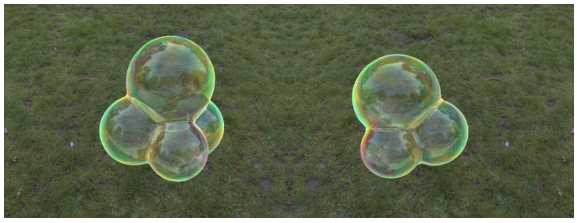

Figure 10: Popping bubbles. Left: a small cluster of bubbles in equilibrium. Right: the cluster rearranges itself into to a new equilibrium after one bubble bursts.

while for cases like cracking an egg and the cherry sauce on a ball, Eqns (6), (8) and (9) are used, as discussed in Section 4.1. We further determine a constant fraction and interaction coefficients between the fluid and the inner gas when simulating air bubbles, as explained in Section 4.2.

These examples demonstrate the significant advantages of using pairwise forces to handle various interactions between multiple phases. Our refined surface tension model gives a practical method of handling multi-phase interfaces—our unified framework is both straightforward, and sufficiently accurate for computer graphics.

6. Conclusions and Future Work

We have proposed a versatile approach for handling multiple interactions at interfaces. Our model is based on pairwise interaction forces. Unlike previous works, it succeeds by adopting a large support radius to eliminate particle deficiency problems. It results in more accurate surface tension forces as explained in Section 3, achieving plausible visual simulation without requiring additional constraints, such as curvature minimization, surface area minimization, energy minimization, or air pressure control. By better understanding the pairwise force model, we avoid the cobweb-like structures seen in some previous work, and produce results closer to those physically observed. Our approach can uniformly handle multiple types of interactions. For the first time in computer graphics, it is possible to simulate bubbles in air as droplets using SPH, providing an alternative method of bubble simulation. Our overall approach is versatile, physically meaningful, and easy to implement. It also conserves momentum and preserves particle clustering. Previous surface tension methods suffer from stiffer time-step restrictions; we achieve more stable simulations by considering more neighboring particles.

We use a fixed number of neighbouring particles, as this leads to simplicity of implementation, convenience in neighborhood search, and consistency in numerical evaluation. It would be interesting to try varying this value to provide error control. The examples presented used physically meaningful equations to tune multiple parameters; previous methods lacked such a simple and unified approach. Pairwise interaction forces actually act at a molecular level, while SPH simulates fluids at a macroscopic level. The proposed model provides an approach to reconcile these two levels. We also observe that the rest particle density can affect the accuracy of interaction forces: the larger the rest particle density becomes, a more important role the interaction forces play.If the rest particle density becomes as high as that of a real liquid at the molecular level, some macroscopic forces such as pressure forces will need to be replaced by interaction forces. To improve the performance of pairwise interaction forces, a larger rest particle density is needed, but this would result in higher computational and memory costs.

While our model can handle various interactions at different interfaces in a unified way, and the relevant coefficients can be determined sequentially, further effort is needed in future to develop easy-to-use artistic control.

Although our approach is capable of capturing adhesion forces and various wetting effects, it cannot handle abrupt, turbulent fluid-solid couplings with large time steps; additional boundary pressure terms are needed in those cases. The work by Akinci et al. [AIA*12] is better suited for such problems.

While it is possible to simulate bubbles in air using our approach, the bubbles behave slightly more rigidly than bubbles simulated by mesh-based methods like those in. [ZQC*14, DBWG15]. Resolving this issue is another direction for future research.

Finally, we note that while increasing the supporting ra-

Figure 11: Wetting effects. Our method is capable of simulating different wetting effects in a simple way.

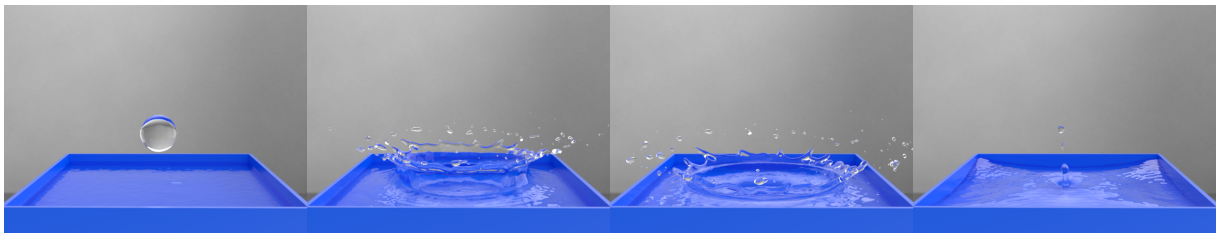

Figure 12: Water crown. A droplet drops into a pool of liquid. Our model produces a plausible water crown and an 'i' pattern.

dius works well in simulating surface tension and adhesion, it cannot fix the particle deficiency issue in SPH. Some form of normalization is required.

Acknowledgements

We would like to thank anonymous reviewers for their constructive comments. This work was supported in part by the Ministry of Education of the People's Republic of China, EU project Dr.Inventor (FP7-ICT-2013.8.1 611383), the U.S. National Science Foundation, and an EPSRC, UK travel grant.

References

[AAT13] AKINCI N., AKINCI G., TESCHNER M.: Versatile Surface Tension and Adhesion for SPH Fluids. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013) 32*, 6 (2013), 182:1–182:8. 2, 3, 4, 6, 8

[AIA*12] AKINCI N., IHMSEN M., AKINCI G., SOLENTHALER B., TESCHNER M.: Versatile Rigid-Fluid Coupling for Incompressible SPH. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 4 (2012), 62:1–62:8. 2, 3, 5, 9

[BB12] BOYD L., BRIDSON R.: MultiFLIP for Energetic Two-Phase Fluid Simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 2 (Apr. 2012), 16:1–16:12. 2

[BBB10] BROCHU T., BATTY C., BRIDSON R.: Matching Fluid Simulation Elements to Surface Geometry and Topology. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010) 29*, 4 (2010), 47:1–47:9. 2

[BDWR12] BUSARYEV O., DEY T. K., WANG H., REN Z.: Animating Bubble Interactions in a Liquid Foam. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 4 (July 2012), 63:1–63:8. 2

[BR15] BO REN YUNTAO JIANG C. L. M. C. L.: A simple approach for bubble modelling from multiphase fluid simulation. *Computer Visual Media 1*, 2 (2015), 171–181. 2

[BT07] BECKER M., TESCHNER M.: Weakly compressible SPH for free surface flows. *In Proceedings of SCA '07* (2007), 209–217. 2, 3, 5

[BTN13] BARECASCO A., TERISSA H., NAA C. F.: Simple free-surface detection in two and three-dimensional SPH solver. *International Symposium on Computational Science* (2013). 7, 8

[BTO*13] BANDARA U., TARTAKOVSKY A., OOSTROM M., PALMER B., GRATE J., ZHANG C.: Smoothed particle hydrodynamics pore-scale simulations of unstable immiscible flow in porous media. *Advances in Water Resources 62* (2013), 356–369. 5, 6

[CBP05] CLAVET S., BEAUDOIN P., POULIN P.: Particle-based Viscoelastic Fluid Simulation. *In Proceedings of SCA '05* (2005), 219–228. 3

[CPPK07] CLEARY P. W., PYO S. H., PRAKASH M., KOO B. K.: Bubbling and Frothing Liquids. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 3 (2007). 6

[DBG14] DA F., BATTY C., GRINSPUN E.: Multimaterial Mesh-Based Surface Tracking. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014) 33*, 4 (2014), 112:1–112:11. 2

[DBWG15] DA F., BATTY C., WOJTAN C., GRINSPUN E.: Double Bubbles Sans Toil and Trouble: Discrete Circulation-Preserving Vortex Sheets for Soap Films and Foams. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2015) 34*, 4 (2015), 149:1–149:9. 2, 7, 9

[Hoe12] HOETZLEIN R.: Fluids V.3. `http://www.rchoetzlein.com/fluids3/`, 2012. 3

[HWZ*14] HE X., WANG H., ZHANG F., WANG H., WANG G., ZHOU K.: Robust Simulation of Small-Scale Thin Features in SPH-based Free Surface Flows. *ACM Transactions on Graphics 34*, 1 (2014), 7:1–7:9. 2, 3, 5, 6

[IBAT11] IHMSEN M., BADER J., AKINCI G., TESCHNER M.: Animation of Air Bubbles with SPH. *Int. Conf. on Computer Graphics Theory and Applications GRAPP* (2011), 225–234. 6

[IOS*14] IHMSEN M., ORTHMANN J., SOLENTHALER B., KOLB A., TESCHNER M.: SPH Fluids in Computer Graphics. *In Eurographics 2014 - State of the Art Reports* (2014), 21–42. 3

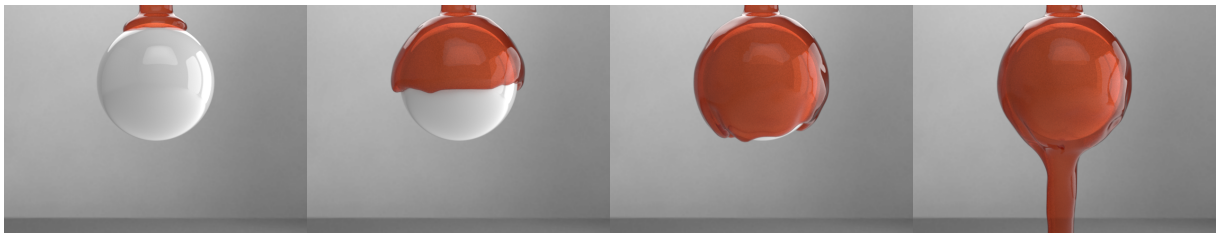

Figure 13: Cherry sauce on a ball. A stream of cherry sauce flows over a ball, sticking to it by the adhesion force between fluid and solid.

[Jak10] JAKOB W.: Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 8

[Kim10] KIM B.: Multi-Phase Fluid Simulations Using Regional Level Sets. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2010) 29*, 6 (Dec. 2010), 175:1–175:8. 2

[KLL*07] KIM B., LIU Y., LLAMAS I., JIAO X., ROSSIGNAC J.: Simulation of Bubbles in Foam With The Volume Control Method. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 3 (July 2007). 2

[KLYK08] KIM J.-M., LEE H.-Y., YOON J.-C., KIM C.-H.: Bubble Alive. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008) 27*, 3 (2008), 48:1–48:4. 2, 6

[LLP11] LIU S., LIU Q., PENG Q.: Realistic simulation of mixing fluids. *The Visual Computer 27*, 3 (Mar. 2011), 241–248. 3

[LMH06] LIU M., MEAKIN P., HUANG H.: Dissipative particle dynamics with attractive and repulsive particle-particle interactions. *Physics of Fluids 18* (2006), 017101. 4

[LSSF06] LOSASSO F., SHINAR T., SELLE A., FEDKIW R.: Multiple Interacting Liquids. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006) 25*, 3 (July 2006), 812–819. 2

[Max90] MAXWELL J.: The Scientific Papers of J.C. Maxwell, Capillary Actions. *Cambridge University Press 2* (1890), 541. 6

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-Based Fluid Simulation for Interactive Application. *In Proceedings of SCA '03* (2003), 154–159. 2, 4

[MEB*12] MISZTAL M., ERLEBEN K., BARGTEIL A., FURSUND J., CHRISTENSEN B., BARENTZEN J., BRIDSON R.: Multiphase Flow of Immiscible Fluids on Unstructured Moving Meshes. *In Proceedings of SCA '12* (2012), 97–106. 2

[Mon89] MONAGHAN J. J.: On the problem of penetration in particle methods. *Journal of Computational Physics 82*, 1 (1989), 1–15. 3

[Mon00] MONAGHAN J. J.: SPH without a tensile instability. *Journal of Computational Physics 159* (2000), 290–311. 2, 5

[Mor00] MORRIS J. P.: Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids 33* (2000), 333–353. 2

[MSKG05] MÜLLER M., SOLENTHALER B., KEISER R., GROSS M.: Particle-Based Fluid-Fluid Interaction. *In Proceedings of SCA '05* (2005), 237–244. 2, 3, 6, 7

[ODAF07] OGER G., DORING M., ALESSANDRINI B., FERRANT P.: An improved SPH method: towards higher order convergence. *Journal of Computational Physics 225* (2007), 1472–1492. 2

[PAKF13] PATKAR S., AANJANEYA M., KARPMAN D., FEDKIW R.: A hybrid Lagrangian-Eulerian formulation for bubble generation and dynamics. *In Proceedings of SCA '13* (2013), 105–114. 6

[Ray64] RAYLEIGH L.: On the theory of surface forces. In: Collected Papers. *New York: Dover 2* (1964), 397–425. 6

[RLY*14] REN B., LI C., YAN X., LIN M. C., BONET J., HU S.-M.: Multiple-fluid SPH Simulation Using a Mixture Model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2014) 33*, 5 (Aug. 2014), 171:1–171:11. 3

[SB12] SCHECHTER H., BRIDSON R.: Ghost SPH for Animating Water. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 4 (2012), 61:1–61:8. 2, 3, 5, 6

[SP08] SOLENTHALER B., PAJAROLA R.: Density contrast SPH Interfaces. *In Proceedings of SCA '08* (2008), 211–218. 3

[SP09] SOLENTHALER B., PAJAROLA R.: Predictive-Corrective Incompressible SPH. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2009) 24*, 3 (2009), 965–972. 5

[SZP07] SOLENTHALER B., ZHANG Y., PAJAROLA R.: Efficient Refinement of Dynamic Point Data. *Eurographics Symposium on Point-Based Graphics* (2007). 7

[TM05] TARTAKOVSKY A., MEAKIN P.: Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E 72*, 2 (2005), 26301. 2, 3, 4

[TTP*14] TARTAKOVSKY A., TRASK N., PAN K., JONES B., PAN W., WILLIAMS J.: Smoothed particle hydrodynamics and its applications for multiphase flow and reactive transport in porous media. *Comput Geosci* (2014), 1–28. 3

[TWGT10] THUEREY N., WOJTAN C., GROSS M., TURK G.: A Multiscale Approach to Mesh-based Surface Tension Flows. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010) 29*, 4 (2010), 48:1–48:10. 2

[YCR*15] YANG T., CHANG J., REN B., LIN M. C., ZHANG J. J., HU S.-M.: Fast Multiple-fluid Simulation Using Helmholtz Free Energy. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2015) 34*, 6 (2015), 201:1–201:11. 3

[You05] YOUNG T.: An Essay on the Cohesion of Fluids. *Phil. Trans. R. Soc. Lond 95* (1805), 65–87. 6

[ZQC*14] ZHU B., QUIGLEY E., CONG M., SOLOMON J., FEWKIW R.: Codimensional Surface Tension Flow on Simplicial Complexes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014) 33*, 4 (2014), 111:1–111:11. 2, 7, 9